**Noname manuscript No.**
(will be inserted by the editor)

# Adaptive Smoothing Algorithms for Nonsmooth Composite Convex Minimization

**Quoc Tran-Dinh**



Received: date / Accepted: date

**Abstract** We propose an adaptive smoothing algorithm based on Nesterov's smoothing technique in [24] for solving "fully" nonsmooth composite convex optimization problems. Our method combines both Nesterov's accelerated proximal gradient scheme and a new homotopy strategy for smoothness parameter. By an appropriate choice of smoothing functions, we develop a new algorithm that has the $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$-worst-case iteration-complexity while preserves the same complexity-per-iteration as in Nesterov's method and allows one to automatically update the smoothness parameter at each iteration. Then, we customize our algorithm to solve four special cases that cover various applications. We also specify our algorithm to solve constrained convex optimization problems and show its convergence guarantee on a primal sequence of iterates. We demonstrate our algorithm through three numerical examples and compare it with other related algorithms.

**Keywords** Nesterov's smoothing technique · accelerated proximal-gradient method · adaptive algorithm · composite convex minimization · nonsmooth convex optimization

**Mathematics Subject Classification (2000)** 90C25 · 90-08

## 1 Introduction

This paper develops new smoothing optimization methods for solving the following "fully" nonsmooth composite convex minimization problem:

$$F^\star := \min_{x\in\mathbb{R}^p}\Big\{F(x) := f(x) + g(x)\Big\}, \tag{1}$$

where $g : \mathbb{R}^p \to \mathbb{R}\cup\{+\infty\}$ is a proper, closed and convex function, and $f : \mathbb{R}^p \to \mathbb{R}\cup\{+\infty\}$ is a convex function defined by the following max-

Quoc Tran-Dinh
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill (UNC), USA.
Email: quoctd@email.unc.edu

structure:

$$f(x) := \max_{u\in\mathbb{R}^n}\Big\{\langle x, Au\rangle - \varphi(u) : u \in \mathcal{U}\Big\}. \tag{2}$$

Here, $\varphi : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ is a proper, closed and convex function, and $\mathcal{U}$ is a nonempty, closed, and convex set in $\mathbb{R}^n$, and $A \in \mathbb{R}^{p\times n}$ is given.

Clearly, any proper, closed and convex function $f$ can be written as (2) using its Fenchel conjugate $f^*$, i.e., $f(x) := \sup\{\langle x, u\rangle - f^*(u) : u \in \mathrm{dom}(f^*)\}$. Hence, the max-structure (2) does not restrict the applicability of the template (1). Moreover, (1) also directly models many practical applications in signal and image processing, machine learning, statistics and data sciences, see, e.g., [4,9,13,15,26,28,32] and the references quoted therein.

While the first term $f$ is nonsmooth, the second term $g$ remains unspecified. On the one hand, we can assume that $g$ is smooth and its gradient is Lipschitz continuous. On the other hand, $g$ can be nonsmooth, but it is equipped with a "tractable" proximity operator defined as follows: $g$ is said to be *tractably proximal* if its proximal operator

$$\mathrm{prox}_g(x) := \mathop{\mathrm{argmin}}_y\big\{g(y) + (1/2)\|y - x\|^2 : y \in \mathrm{dom}(g)\big\}, \tag{3}$$

can be computed "efficiently" (e.g., by a closed form or by polynomial time algorithms). In general, computing $\mathrm{prox}_g$ requires to solve the strongly convex problem (3), but in many cases, this operator can be obtained in a closed form or by a low-cost polynomial algorithm. Examples of such convex functions can be found in the literature including [3,15,28].

Solving nonsmooth convex optimization problems remains challenging, especially when none of the two nonsmooth terms $f$ and $g$ is equipped with a tractable proximity operator. Existing nonsmooth convex optimization approaches such as subgradient-type descent algorithms, dual averaging strategies, bundle-level techniques or derivative-free methods are often used to solve general nonsmooth convex problems. However, these methods suffer a slow convergence rate (resp., $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ - worst-case iteration-complexity). In addition, they are sensitive to the algorithmic parameters such as stepsizes [22].

In his pioneering work [24], Nesterov shown that one can solve the nonsmooth structured convex minimization problem (1) within $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ iterations. This method combines a proximity smoothing technique and Nesterov's accelerated gradient scheme [21] to achieve the optimal worst-case iteration-complexity, which is much better than the $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$-worst-case iteration complexity in nonsmooth optimization methods.

Motivated by [24], Nesterov and many other researchers have proposed different algorithms using such a proximity smoothing method to solver other problems, to improve Nesterov's original algorithm or customize his algorithm to specific applications, see, e.g., [2,6,7,14,17,19,20,23,25,33]. In [5], Beck and Teboulle generalized Nesterov's smoothing technique to a generic framework, where they discussed the advantages and disadvantages of smoothing techniques. In addition, they also illustrated the numerical efficiency between smoothing techniques and proximal-type methods. In [1,27], the authors studied smoothing techniques for the sum of three convex functions, where one

term is Lipschitz gradient, while the others are nonsmooth. In [11], a variable smoothing method was proposed, which possesses the $\mathcal{O}\left(\frac{\ln(k)}{k}\right)$-convergence rate. This convergence rate is worse than the one in [24]. However, as a compensation, the smoothness parameter is updated at each iteration. In addition, their method uses special quadratic proximity functions, while smooths both $f$ and $g$ under their Lipschitz continuity assumption.

In [23], Nesterov introduced an excessive gap technique, which requires both primal and dual schemes using two smoothness parameters. It symmetrically updates one parameter at each iteration. Nevertheless, this method uses different assumptions than our method. Other primal-dual methods studied in, e.g., [12,16] use double smoothing techniques to solve (1), but only achieve $\mathcal{O}\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\varepsilon}\right)\right)$-worst-case iteration-complexity.

Our approach in this paper is also based on Nesterov's smoothing technique in [24]. To clarify the differences between our method and [23,24], let us first briefly present Nesterov's smoothing technique in [24] applying to (1).

Recall that a convex function $b_{\mathcal{U}} : \mathbb{R}^n \to \mathbb{R}$ is a proximity function of $\mathcal{U}$ if it is continuous, and strongly convex with the convexity parameter $\mu_b > 0$ and $\mathcal{U} \subseteq \mathrm{dom}(b_{\mathcal{U}})$. We define

$$\bar{u}^c := \underset{u}{\mathrm{argmin}}\left\{b_{\mathcal{U}}(u) : u \in \mathcal{U}\right\} \quad \text{and} \quad D_{\mathcal{U}} := \sup_{u}\left\{b_{\mathcal{U}}(u) : u \in \mathcal{U}\right\} \in [0, +\infty).$$

Here, $\bar{u}^c$ and $D_{\mathcal{U}}$ are called the prox-center and prox-diameter of $\mathcal{U}$ w.r.t. $b_{\mathcal{U}}$, respectively. Without loss of generality, we can assume that $b_{\mathcal{U}}(\bar{u}^c) = 0$ and $\mu_b = 1$. Otherwise, we just rescale and shift it.

As shown in [24], given $\gamma > 0$ and $b_{\mathcal{U}}$, we can approximate $f$ by $f_\gamma$ as

$$f_\gamma(x) := \max_{u}\left\{\langle x, Au\rangle - \varphi(u) - \gamma b_{\mathcal{U}}(u) : u \in \mathcal{U}\right\}, \tag{4}$$

where $\gamma$ is called a smoothness parameter. Since $f_\gamma$ is smooth and has Lipschitz gradient, one can apply accelerated proximal gradient methods [4,26] to minimize the sum $f_\gamma(\cdot) + g(\cdot)$. Using such methods, we can eventually guarantee

$$F(x^k) - F^\star \le \min_{\gamma>0}\left\{\frac{2\|A\|^2R_0^2}{\gamma(k+1)^2} + \gamma D_{\mathcal{U}}\right\} = \frac{2\sqrt{2}\|A\|R_0\sqrt{D_{\mathcal{U}}}}{(k+1)}, \tag{5}$$

where $\left\{x^k\right\}$ is the underlying sequence generated by the accelerated proximal-gradient method, see [24], and $R_0 := \|x^0 - x^\star\|$. To achieve an $\varepsilon$-solution $x^k$ such that $F(x^k) - F^\star \le \varepsilon$, we set $\gamma \equiv \gamma^* := \frac{\varepsilon}{2D_{\mathcal{U}}}$ at its optimal value. Hence, the algorithm requires at most $k_{\max} := \left\lfloor 2\sqrt{2}\|A\|R_0\sqrt{D_{\mathcal{U}}}\varepsilon^{-1}\right\rfloor$ iterations.

*Our approach:* The original smoothing algorithm in [24] has three computational disadvantages even with the optimal choice $\gamma^* := \frac{\varepsilon}{2D_{\mathcal{U}}}$ of $\gamma$.

(a) It requires the prox-diameter $D_{\mathcal{U}}$ of $\mathcal{U}$ to determine $\gamma^*$, which may be expensive to estimate when $\mathcal{U}$ is complicated.
(b) If $\varepsilon$ is small and $D_{\mathcal{U}}$ is large, then $\gamma^*$ is small, and hence, the strong convexity parameter of (4) is small. Algorithms for solving (4) have slow convergence speed.

(c) The Lipschitz constant of $\nabla f_\gamma$ is $\gamma^{-1}\|A\|^2 = \|A\|^2 D_{\mathcal{U}}\varepsilon^{-1}$, which is large. This leads to a small step-size of $\varepsilon/(\|A\|^2 D_{\mathcal{U}})$ in the accelerated proximal-gradient algorithm and hence, can have a slow convergence.

Our approach is briefly presented as follows. We first choose a smooth proximity function $b_{\mathcal{U}}$ instead of a general one. We assume that $\nabla b_{\mathcal{U}}$ is $L_b$-Lipschitz continuous with the Lipschitz constant $L_b \geq \mu_b = 1$. Then, we define $f_\gamma(x)$ as in (4), which is a smoothed approximation to $f$ as above.

We design a smoothing accelerated proximal-gradient algorithm that can updates $\gamma$ from $\gamma_k$ to $\gamma_{k+1}$ at each iteration so that $\gamma_{k+1} < \gamma_k$ by performing only *one* accelerated proximal-gradient step [4,26] to minimize the sum $F_{\gamma_{k+1}} := f_{\gamma_{k+1}} + g$ for each value $\gamma_{k+1}$ of $\gamma$. We prove that the sequence of the objective residuals, $\{F(x^k) - F^\star\}$, converges to zero up to the $\mathcal{O}\left(\frac{1}{k}\right)$-rate.

*Our contributions:* Our main contributions can be summarized as follows:

(a) We propose using a smooth proximity function to smooth the max-structure objective function $f$ in (2), and develop a new smoothing algorithm, Algorithm 1, based on the accelerated proximal-gradient method to adaptively update the smoothness parameter in a heuristic-free fashion.
(b) We prove up to the $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$-worst-case iteration-complexity for our algorithm as in [24] to achieve an $\varepsilon$-solution, i.e., $F(x^k) - F^\star \leq \varepsilon$. Especially, with the quadratic proximity function $b_{\mathcal{U}}(\cdot) := (1/2)\|\cdot - \bar{u}^c\|^2$, our algorithm achieve exactly the $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$-worst-case iteration-complexity as in [24].
(c) We customize our algorithm to handle four important special cases that have a great practical impact in many applications.
(d) We specify our algorithm to solve constrained convex minimization problems, and propose an averaging scheme to recover an approximate primal solution with a rigorous convergence guarantee.

From a practical point of view, we believe that the proposed algorithm can overcome three disadvantages mentioned previously in the original smoothing algorithm in [24]. However, our condition $L_b = 1$ on the choice of proximity functions may lead to some limitation of the proposed algorithm for exploiting further the structures of the constrained set $\mathcal{U}$. Fortunately, we can identify several important settings in Section 4, where we can eliminate this disadvantage. Such classes of problems cover several applications in image processing, compressive sensing, and monotropic programming [3,15,28,34].

*Paper organization:* The rest of this paper is organized as follows. Section 2 briefly discusses our smoothing technique. Section 3 presents our main algorithm, Algorithm 1, and proves its convergence guarantee. Section 4 handles four special but important cases of (1). Section 5 specializes our algorithm to solve constrained convex minimization problems. Preliminarily numerical examples are given in Section 6. For clarity of presentation, we move the long and technical proofs to the appendix.

*Notation and terminology:* We work on the real spaces $\mathbb{R}^p$ and $\mathbb{R}^n$, equipped with the standard inner product $\langle\cdot,\cdot\rangle$ and the Euclidean $\ell_2$-norm $\|\cdot\|$. Given a proper, closed, and convex function $g$, we use $\mathrm{dom}(g)$ and $\partial g(x)$ to denote

its domain and its subdifferential at $x$, respectively. If $g$ is differentiable, then $\nabla g(x)$ stands for its gradient at $x$.

We denote $f^*(s) := \sup\{\langle s, x\rangle - f(x) : x \in \mathrm{dom}(f)\}$, the Fenchel conjugate of $f$. For a given set $\mathcal{X}$, $\delta_{\mathcal{X}}(x) := 0$ if $x \in \mathcal{X}$ and $\delta_{\mathcal{X}}(x) := +\infty$, otherwise, defines the indicator function of $\mathcal{X}$. For a smooth function $f$, we say that $f$ is $L_f$-smooth if for any $x, \tilde{x} \in \mathrm{dom}(f)$, we have $\|\nabla f(x) - \nabla f(\tilde{x})\| \le L_f\|x - \tilde{x}\|$, where $L(f) := L_f \in [0, \infty)$. We denote by $\mathcal{F}_L^{1,1}$ the class of all $L_f$-smooth and convex functions $f$. We also use $\mu_f \equiv \mu(f)$ for the strong convexity parameter of a convex function $f$. For a given symmetric matrix $X$, $\lambda_{\min}(X)$ and $\lambda_{\max}(X)$ denote its smallest and largest eigenvalues of $X$, respectively; and $\mathrm{cond}(X)$ is the condition number of $X$. Given a nonempty, closed and convex set $\mathcal{X}$, $\mathrm{dist}\,(x, \mathcal{X})$ denotes the distance from $x$ to $\mathcal{X}$.

## 2 Smoothing techniques via smooth proximity functions

Let $b_{\mathcal{U}}$ be a prox-function of the nonempty, closed and convex set $\mathcal{U}$ with the strong convexity parameter $\mu_b = 1$. In addition, $b_{\mathcal{U}}$ is smooth on $\mathcal{U}$, and its gradient $\nabla b_{\mathcal{U}}$ is Lipschitz continuous with the Lipschitz constant $L_b \ge \mu_b = 1$. In this case, $b_{\mathcal{U}}$ is said to be $L_b$-smooth. As a default example, $b_{\mathcal{U}}(\cdot) := (1/2)\|\cdot - \bar{u}^c\|^2$ for fixed $\bar{u}^c \in \mathcal{U}$ satisfies our assumptions with $L_b = \mu_b = 1$. Let $\bar{u}^c$ be the $b$-prox-center point of $\mathcal{U}$, i.e., $\bar{u}^c := \mathrm{argmin}_u\{b_{\mathcal{U}}(u) : u \in \mathcal{U}\}$. Without loss of generality, we can assume that $b_{\mathcal{U}}(\bar{u}^c) = 0$. Otherwise, we consider $\bar{b}_{\mathcal{U}}(u) := b_{\mathcal{U}}(u) - b_{\mathcal{U}}(\bar{u}^c)$.

Given a convex function $\varphi^*(z) := \max_u\{\langle z, u\rangle - \varphi(u) : u \in \mathcal{U}\}$, we define a smoothed approximation of$\varphi^*$ as

$$\varphi_\gamma^*(z) := \max_{u\in\mathcal{U}}\{\langle z, u\rangle - \varphi(u) - \gamma b_{\mathcal{U}}(u)\}, \tag{6}$$

where $\gamma > 0$ is a smoothness parameter. We note that $\varphi^*$ is not a Fenchel conjugate of $\varphi$ unless $\mathcal{U} = \mathrm{dom}(\varphi)$. We denote by $u_\gamma^*(x)$ the unique optimal solution of the strongly concave maximization problem (6), i.e.:

$$u_\gamma^*(z) \in \mathrm{argmax}_u\{\langle z, u\rangle - \varphi(u) - \gamma b_{\mathcal{U}}(u) : u \in \mathcal{U}\}. \tag{7}$$

We also define $D_{\mathcal{U}} := \sup_u\{b_{\mathcal{U}}(u) : u \in \mathcal{U} \cap \mathrm{dom}(\varphi)\}$ the $b$-prox diameter of $\mathcal{U}$. If $\mathcal{U}$ or $\mathrm{dom}(\varphi)$ is bounded, then $D_{\mathcal{U}} \in [0, +\infty)$.

Associated with $\varphi_\gamma^*$, we consider a smoothed function for $f$ in (2) as

$$f_\gamma(x) := \varphi_\gamma^*(A^\top x) = \max_u\left\{\langle A^\top x, u\rangle - \varphi(u) - \gamma b_{\mathcal{U}}(u) : u \in \mathcal{U}\right\}. \tag{8}$$

Then, the following lemma summaries the properties of the smoothed function $\varphi_\gamma^*$ defined by (6) and $f_\gamma$ defined by (8), whose proof can be found in [31].

**Lemma 1** *The function $\varphi_\gamma^*$ defined by* (6) *is convex and smooth. Its gradient is given by $\nabla\varphi_\gamma^*(z) := u_\gamma^*(z)$ which is Lipschitz continuous with the Lipschitz constant $L_{\varphi_\gamma^*} := \gamma^{-1}$. Consequently, for any $z, \bar{z} \in \mathbb{R}^n$, we have*

$$\frac{\gamma}{2}\|u_\gamma^*(z) - u_\gamma^*(\bar{z})\|^2 \le \varphi_\gamma^*(z) - \varphi_\gamma^*(\bar{z}) - \langle\nabla\varphi_\gamma^*(z), z - \bar{z}\rangle \le \frac{1}{2\gamma}\|z - \bar{z}\|^2. \tag{9}$$

*For fixed $z \in \mathbb{R}^n$, $\varphi_\gamma^*(z)$ is convex w.r.t. $\gamma \in \mathbb{R}_{++}$, and*

$$\varphi_\gamma^*(z) - (\hat{\gamma} - \gamma) b_{\mathcal{U}}(u_\gamma^*(z)) \le \varphi_{\hat{\gamma}}^*(z), \quad \forall \gamma, \hat{\gamma} \in \mathbb{R}_{++}. \tag{10}$$

*As a consequence, $f_\gamma$ defined by (8) is convex and smooth. Its gradient is given by $\nabla f_\gamma(x) = Au_\gamma^*(A^\top x)$, which is Lipschitz continuous with the Lipschitz constant $L_{f_\gamma} := \gamma^{-1}\|A\|^2$. In addition, we also have*

$$f_\gamma(x) \le f(x) \le f_\gamma(x) + \gamma D_{\mathcal{U}}, \quad \forall x \in \mathbb{R}^p. \tag{11}$$

We emphasize that Lemma 1 provides key properties to analyze the complexity of our algorithm in the next setions.

## 3 The adaptive smoothing algorithm and its convergence

Associated with (1), we consider its smoothed composite convex problem as

$$F_\gamma^\star := \min_{x \in \mathbb{R}^p} \left\{ F_\gamma(x) := f_\gamma(x) + g(x) \right\}. \tag{12}$$

Similar to [24], the main step of Nesterov's accelerated proximal-gradient scheme [4,26] applied to the smoothed problem (12) is expressed as follows:

$$\begin{aligned} x^{k+1} &:= \mathrm{prox}_{\beta g}\left(\hat{x}^k - \beta \nabla f_\gamma(\hat{x}^k)\right) \\ &\equiv \arg\min_{x \in \mathbb{R}^p} \left\{ g(x) + \frac{1}{2\beta}\left\| x - \left(\hat{x}^k - \beta A u_\gamma^*(A^\top \hat{x}^k)\right)\right\|^2 \right\}, \end{aligned} \tag{13}$$

where $\hat{x}^k$ is given, and $\beta > 0$ is a given step size, which will be chosen later.

The following lemma provides a descent property of the proximal-gradient step (13), whose proof can be found in Appendix A.1.

**Lemma 2** *Let $x^{k+1}$ be generated by (13). Then, for any $x \in \mathbb{R}^p$, we have*

$$F_\gamma(x^{k+1}) \le \hat{\ell}_\gamma^k(x) + \tfrac{1}{\beta}\langle x^{k+1} - \hat{x}^k, x - \hat{x}^k\rangle - \tfrac{1}{2}\left(\tfrac{2}{\beta} - \tfrac{\|A\|^2}{\gamma}\right)\|\hat{x}^k - x^{k+1}\|^2, \tag{14}$$

*where*

$$\begin{aligned} \hat{\ell}_\gamma^k(x) &:= f_\gamma(\hat{x}^k) + \langle \nabla f_\gamma(\hat{x}^k), x - \hat{x}^k\rangle + g(x) \\ &\le F_\gamma(x) - \tfrac{\gamma}{2}\|u_\gamma^*(A^\top x) - u_\gamma^*(A^\top \hat{x}^k)\|^2. \end{aligned} \tag{15}$$

We now adopt the accelerated proximal-gradient scheme (FISTA) in [4] to solve (12) using an adaptive step-size $\beta_{k+1} := \frac{\gamma_{k+1}}{\|A\|^2}$, which becomes

$$\begin{cases} \hat{x}^k &:= (1-\tau_k)x^k + \tau_k \tilde{x}^k \\ x^{k+1} &:= \mathrm{prox}_{\beta_{k+1} g}\left(\hat{x}^k - \beta_{k+1}\nabla f_{\gamma_{k+1}}(\hat{x}^k)\right) \\ \tilde{x}^{k+1} &:= \tilde{x}^k - \frac{1}{\tau_k}(\hat{x}^k - x^{k+1}), \end{cases} \tag{16}$$

where $\gamma_{k+1} > 0$ is the smoothness parameter, and $\tau_k \in (0, 1]$.

By letting $t_k := \frac{1}{\tau_k}$, we can eliminate $\tilde{x}^k$ in (16) to obtain a compact version

$$\begin{cases} x^{k+1} &:= \mathrm{prox}_{\beta_{k+1} g}\left(\hat{x}^k - \beta_{k+1}\nabla f_{\gamma_{k+1}}(\hat{x}^k)\right) \\ \hat{x}^{k+1} &:= x^{k+1} + \frac{t_k - 1}{t_{k+1}}(\hat{x}^k - x^k). \end{cases} \tag{17}$$

The following lemma provides a key estimate to prove the convergence of the scheme (16) (or (17)), whose proof can be found in Appendix A.2.

**Lemma 3** *Let $\{(x^k, \tilde{x}^k, \tau_k, \gamma_k)\}$ be the sequence generated by (16). Then*

$$F_{\gamma_{k+1}}(x^{k+1}) \le (1-\tau_k)F_{\gamma_k}(x^k)+\tau_k F(x)+\frac{\|A\|^2\tau_k^2}{2\gamma_{k+1}}\left[\|\tilde{x}^k-x\|^2-\|\tilde{x}^{k+1}-x\|^2\right]-R_k, \tag{18}$$

*for any $x \in \mathbb{R}^p$ and $R_k$ is defined by*

$$\begin{aligned} R_k &:= \tau_k\gamma_{k+1}b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)) - (1-\tau_k)(\gamma_k-\gamma_{k+1})b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top x^k)) \\ &\quad + \tfrac{(1-\tau_k)\gamma_{k+1}}{2}\|u^*_{\gamma_{k+1}}(A^\top x^k)-u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)\|^2. \end{aligned} \tag{19}$$

*Moreover, the quantity $R_k$ is bounded from below by*

$$R_k \ge \tfrac{1}{2}(1-\tau_k)\left[\gamma_{k+1}\tau_k - L_b(\gamma_k-\gamma_{k+1})\right]b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top x^k)). \tag{20}$$

Next, we show *one* possibility for updating $\tau_k$ and $\gamma_k$, and provide an upper bound for $F_{\gamma_k}(x^k) - F^\star$. The proof of this lemma is moved to Appendix A.3.

**Lemma 4** *Let us choose $\tilde{x}^0 := x^0 \in \mathrm{dom}(F)$, $\gamma_1 > 0$, and an arbitrary constant $\bar{c} \ge 1$. If the parameters $\tau_k$ and $\gamma_k$ are updated by*

$$\tau_k := \frac{1}{k+\bar{c}} \quad \text{and} \quad \gamma_{k+1} := \frac{\gamma_1\bar{c}}{k+\bar{c}}, \tag{21}$$

*then the quantity $R_k$ defined by (19) and $\{(\tau_k, \gamma_k)\}$ satisfy*

$$\frac{\gamma_{k+1}}{\tau_k^2}R_k \ge -\frac{\gamma_1^2\bar{c}^2\left[(L_b-1)(k+\bar{c})+1\right]}{(k+\bar{c})^2}D_{\mathcal{U}} \quad \text{and} \quad \frac{(1-\tau_k)\gamma_{k+1}}{\tau_k^2} = \frac{\gamma_k}{\tau_{k-1}^2}. \tag{22}$$

*Moreover, the following estimate holds*

$$F_{\gamma_{k+1}}(x^{k+1})-F^\star \le \frac{\tau_k^2}{\gamma_{k+1}}\left[\frac{(1-\tau_0)\gamma_1}{\tau_0^2}(F_{\gamma_0}(x^0)-F^\star)+\frac{\|A\|^2}{2}\|x^0-x^\star\|^2+S_kD_{\mathcal{U}}\right], \tag{23}$$

*where*

$$S_k := \gamma_1^2\bar{c}^2\textstyle\sum_{i=0}^k\left[\frac{(L_b-1)}{(i+\bar{c})}+\frac{1}{(i+\bar{c})^2}\right] \le \gamma_1^2\bar{c}^2(L_b-1)\left(\ln(k+\bar{c})+1\right)+\gamma_1^2(\bar{c}+1). \tag{24}$$

*In particular, if we choose $b_{\mathcal{U}}$ such that $L_b = 1$, then $S_k \le \gamma_1^2(\bar{c}+1)$.*

By (21), the second line of (17) reduces to $\hat{x}^{k+1} := x^{k+1} + \left(\frac{k+\bar{c}-1}{k+\bar{c}+1}\right)(x^{k+1} - x^k)$. Using this step into (17) and combining the result with the update rule (21), we can present our algorithm for solving (1) as in Algorithm 1.

The following theorem proves the convergence of Algorithm 1 and estimates its worst-case iteration-complexity.

**Theorem 1** *Let $\{x^k\}$ be the sequence generated by Algorithm 1 using $\bar{c} = 1$. Then, for $k \ge 1$, we have*

$$F(x^k) - F^\star \le \frac{\|A\|^2\|x^0-x^\star\|^2}{2\gamma_1 k} + \frac{3\gamma_1 D_{\mathcal{U}}}{k} + \frac{\gamma_1(L_b-1)\left(\ln(k)+1\right)D_{\mathcal{U}}}{k}. \tag{25}$$

**Algorithm 1** (*Adaptive Smoothing Proximal-Gradient Algorithm*)

**Initialization:**

1: Choose $\gamma_1 > 0$, $\bar{c} \geq 1$ and $x^0 \in \mathbb{R}^p$. Set $\hat{x}^0 := x^0$.

**Iteration: For** $k = 0$ **to** $k_{\max}$, **perform:**

2: Solve the following strongly concave maximization subproblem

$$\hat{u}^*_{\gamma_{k+1}}(\hat{x}^k) := \mathop{\mathrm{argmax}}_{u\in\mathcal{U}}\Big\{\langle \hat{x}^k, Au\rangle - \varphi(u) - \gamma_{k+1}b_{\mathcal{U}}(u)\Big\}.$$

3: Perform the following proximal-gradient step with $\beta_{k+1} := \frac{\gamma_{k+1}}{\|A\|^2}$:

$$x^{k+1} := \mathrm{prox}_{\beta_{k+1}g}\left(\hat{x}^k - \beta_{k+1}A\hat{u}^*_{\gamma_{k+1}}(\hat{x}^k)\right).$$

4: Update $\hat{x}^{k+1} := x^{k+1} + \left(\frac{k+\bar{c}-1}{k+\bar{c}+1}\right)(x^{k+1} - x^k)$.

5: Compute $\gamma_{k+2} := \frac{\bar{c}\gamma_1}{k+\bar{c}+1}$.

**End for**

*If $b_{\mathcal{U}}$ is chosen so that $L_b = 1$ (e.g., $b_{\mathcal{U}}(\cdot) := \frac{1}{2}\|\cdot - \bar{u}^c\|^2$), then* (25) *reduces to*

$$F(x^k) - F^\star \leq \frac{\|A\|^2\|x^0 - x^\star\|^2}{2\gamma_1 k} + \frac{3\gamma_1 D_{\mathcal{U}}}{k}, \quad (\forall k \geq 1). \tag{26}$$

*Consequently, if we set $\gamma_1 := \frac{R_0\|A\|}{\sqrt{6D_{\mathcal{U}}}}$, which is independent of $k$, then*

$$F(x^k) - F^\star \leq \frac{R_0\|A\|\sqrt{6D_{\mathcal{U}}}}{k} \quad (\forall k \geq 1), \tag{27}$$

*where $R_0 := \|x^0 - x^\star\|$.*

*In this case, the worst-case iteration-complexity of Algorithm 1 to achieve an $\varepsilon$-solution $x^k$ to* (1) *such that $F(x^k) - F^\star \leq \varepsilon$ is $k_{\max} := \mathcal{O}\Big(\frac{R_0\|A\|\sqrt{D_{\mathcal{U}}}}{\varepsilon}\Big)$.*

*Proof* From (21), $\bar{c} = 1$ we have $\frac{\tau^2_{k-1}}{\gamma_k} = \frac{(k+\bar{c}-1)}{\bar{c}\gamma_1(k+\bar{c}-1)^2} = \frac{1}{\gamma_1 k}$. Using this bound and $S_{k-1} \leq \gamma_1^2(L_b - 1)\left[\ln(k) + 1\right] + 2\gamma_1^2$ into (23) we get

$$\begin{aligned} F_{\gamma_k}(x^k) - F^\star \leq \frac{1}{\gamma_1 k}\left[\frac{\|A\|^2}{2}\|x^0 - x^\star\|^2 + \frac{\gamma_1(1-\tau_0)}{\tau_0^2}[F_{\gamma_0}(x^0) - F^\star]\right] \\ + \frac{\left(\gamma_1(L_b - 1)\left[\ln(k) + 1\right] + 2\gamma_1\right)D_{\mathcal{U}}}{k}. \end{aligned}$$

Since $F(x^k) - F_{\gamma_k}(x^k) \leq \gamma_k D_{\mathcal{U}}$ due to (11), and $\gamma_k = \frac{\gamma_1\bar{c}}{k+\bar{c}-1} = \frac{\gamma_1}{k}$. Substituting this inequality into the last estimate, and using $\tau_0 = \frac{1}{\bar{c}} = 1$, we obtain (25).

If we choose $b_{\mathcal{U}}$ such that $L_b = 1$, e.g., $b_{\mathcal{U}}(\cdot) := (1/2)\|\cdot - \bar{u}^c\|^2$, then $S_k \leq 2\gamma_1^2$ as shown in (24). Using this, it follows from (25) that $F(x^k) - F^\star \leq \frac{\|A\|}{2\gamma_1 k}R_0^2 + \frac{3\gamma_1}{k+1}D_{\mathcal{U}}$. By minimizing the right hand side of this estimate w.r.t $\gamma_1 > 0$, we have $\gamma_1 := \frac{R_0\|A\|}{\sqrt{6D_{\mathcal{U}}}}$ and hence, $F(x^k) - F^\star \leq \frac{R_0\|A\|\sqrt{6D_{\mathcal{U}}}}{k}$, which is exactly (27). The last statement is a direct consequence of (27). □

For general prox-function $b_{\mathcal{U}}$ with $L_b > 1$, Theorem 1 shows that the convergence rate of Algorithm 1 is $\mathcal{O}\left(\frac{\ln(k)}{k}\right)$, which is similar to [11]. However, when $L_b$ is close to 1, the last term in (25) is better than [11, Theorem 1].

*Remark 1* Let $b_{\mathcal{U}}(\cdot) := (1/2)\|\cdot - \bar{u}^c\|^2$. Then, (27) shows that the number of maximum iterations in Algorithm 1 is $k_{\max} := \left\lfloor \frac{R_0\|A\|\sqrt{6D_{\mathcal{U}}}}{\varepsilon} \right\rfloor$, which is the same, $k_{\max} := \left\lfloor \frac{2\sqrt{2}\|A\|R_0\sqrt{D_{\mathcal{U}}}}{\varepsilon} \right\rfloor$, as in (5) (with different factors, $\sqrt{6}$ and $2\sqrt{2}$).

## 4 Exploiting structures for special cases

For general smooth proximity function $b_{\mathcal{U}}$ with $L_b > 1$, we can achieve the $\mathcal{O}\left(\frac{(L_b-1)\ln(k)}{k}\right)$ convergence rate. When $L_b = 1$, we obtain exactly the $\mathcal{O}\left(\frac{1}{k}\right)$ rate as in [24]. In this section, we consider three special cases of (1) where we use the quadratic proximity function $b_{\mathcal{U}}(\cdot) := (1/2)\|\cdot - \bar{u}^c\|^2$. Then, we specify Algorithm 1 for the $L_g$-smooth objective function $g$ in (1).

### 4.1 Fenchel conjugate

Let $f^*$ be the Fenchel conjugate of $f$. We can write $f$ in the form of (2) as

$$f(x) = \max_u \{\langle x, u\rangle - f^*(u) : u \in \mathrm{dom}(f^*)\}.$$

We can smooth $f$ by using $b_{\mathcal{U}}(u) := (1/2)\|u\|_2^2$ as

$$f_\gamma(x) := \max_{u\in\mathrm{dom}(f^*)} \left\{\langle x, u\rangle - f^*(u) - (\gamma/2)\|u\|_2^2\right\} = \frac{\|x\|^2}{2\gamma} - {}^{\gamma^{-1}}f^*(\gamma^{-1}x),$$

where ${}^{\beta}h$ is the Moreau envelope of a convex function $h$ with a parameter $\beta$ [3]. In this case, $u_\gamma^*(x) = \mathrm{prox}_{\gamma^{-1}f^*}(\gamma^{-1}x) = \gamma^{-1}(x - \mathrm{prox}_{\gamma f}(x))$. Hence, $\nabla f_\gamma(x) = \gamma^{-1}(x - \mathrm{prox}_{\gamma f}(x))$. The main step, Step 3, of Algorithm 1 becomes

$$x^{k+1} = \mathrm{prox}_{\gamma_{k+1}g}\big(\mathrm{prox}_{\gamma_{k+1}f}(\hat{x}^k)\big).$$

Hence, Algorithm 1 can be applied to solve (1) using the proximal operator of $f$ and $g$. The worst-case complexity bound in Theorem 1 becomes $\mathcal{O}\left(\frac{D_{\mathrm{dom}(f^*)}R_0}{\varepsilon}\right)$, where $D_{\mathrm{dom}(f^*)} := \max_{u\in\mathrm{dom}(f^*)} \|u\|$ is the diameter of $\mathrm{dom}(f^*)$.

### 4.2 Composite convex minimization with linear operator

We consider the following composite convex problem with a linear operator that covers many important applications in practice, see, e.g., [1,3,15]:

$$F^\star := \min_{x\in\mathbb{R}^p} \{F(x) := f(Ax) + g(x)\}, \tag{28}$$

where $f$ and $g$ are two proper, closed and convex functions, and $A$ is a linear operator from $\mathbb{R}^p$ to $\mathbb{R}^n$.

We first write $f(Ax) := \max_u \{\langle Ax, u\rangle - f^*(u) : u \in \mathrm{dom}(f^*)\}$. Next, we choose a quadratic smoothing proximity function $b_{\mathcal{U}}(u) := (1/2)\|u - \bar{u}^c\|^2$ for

fixed $\bar{u}^c \in \mathrm{dom}(f^*)$, and define $\mathcal{U} := \mathrm{dom}(f^*)$. Using this smoothing prox-function, we obtain a smoothed approximation of $f(Ax)$ as follows:

$$f_\gamma(Ax) := \max_u \left\{ \langle Ax, u\rangle - f^*(u) - (\gamma/2)\|u - \bar{u}^c\|^2 : u \in \mathrm{dom}(f^*) \right\}.$$

In this case, we can compute $u^*_\gamma(Ax) = \mathrm{prox}_{\gamma^{-1}f^*}\left(\bar{u}^c + \gamma^{-1}Ax\right)$ by using the proximal operator of $f^*$. By Fenchel-Moreau's decomposition $\mathrm{prox}_{\gamma^{-1}f^*}(\gamma^{-1}v) = \gamma^{-1}(v - \mathrm{prox}_{\gamma f}(\gamma v))$ as above, we can compute $\mathrm{prox}_{\gamma^{-1}f^*}$ using the proximal operator of $f$. In this case, we can specify the proximal-gradient step (13) as

$$\begin{cases} \hat{u}^*_k &:= \mathrm{prox}_{\gamma^{-1}_{k+1}f^*}\left(\bar{u}^c + \gamma^{-1}_{k+1}A\hat{x}^k\right) \\ &= \bar{u}^c + \gamma^{-1}_{k+1}\left(A\hat{x}^k - \mathrm{prox}_{\gamma_{k+1}f}\left(\gamma_{k+1}\bar{u}^c + A\hat{x}^k\right)\right) \\ x^{k+1} &:= \mathrm{prox}_{\beta_{k+1}g}\left(\hat{x}^k - \beta_{k+1}A^\top\hat{u}^*_k\right), \end{cases}$$

where $\beta_{k+1} := \gamma_{k+1}\|A\|^{-2}$. Using this proximal gradient step in Algorithm 1, we still obtain the complexity as in Theorem 1, which is $\mathcal{O}\Big(\frac{\|x^0 - x^\star\|\|A\|\sqrt{D_{\mathcal{U}}}}{\varepsilon}\Big)$, where the domain $\mathcal{U} := \mathrm{dom}(f^*)$ of $f^*$ is assumed to be bounded.

### 4.3 The decomposable structure

The function $\varphi$ and the set $\mathcal{U}$ in (2) are said to be *decomposable* if they can be represented as follows:

$$\varphi(u) := \sum_{i=1}^m \varphi_i(u_i), \quad \text{and} \quad \mathcal{U} := \mathcal{U}_1 \times \cdots \times \mathcal{U}_m, \tag{29}$$

where $m \geq 2$, $u_i \in \mathbb{R}^{n_i}$, $\mathcal{U}_i \subseteq \mathbb{R}^{n_i}$ and $\sum_{i=1}^m n_i = n$. In this case, we also say that problem (1) is *decomposable*.

The structure (29) naturally arises in linear programming and monotropic programming. In addition, many nondecomposable problems such as consensus optimization, empirical loss optimization, conic programming and geometric programming can also be reformulated into (1) with the structure (29). The decomposable structure (29) immediately supports parallel and distributed computation. Exploiting this structure, one can design new parallel and distributed optimization algorithms using the same approach as in Algorithm 1 for solving (1), see, e.g., [10,13,29,30].

Under the structure (29), we choose a decomposable smoothing function $b_{\mathcal{U}}(u) := \sum_{i=1}^m b_{\mathcal{U}_i}(u_i)$, where $b_{\mathcal{U}_i}$ is the prox-function of $\mathcal{U}_i$ for $i = 1, \cdots, m$. The smoothed function $f_\gamma$ for $f$ is decomposable, and is represented as follows:

$$f_\gamma(x) := \sum_{i=1}^m \left\{ f^i_\gamma(x) := \max_{u_i \in \mathcal{U}_i} \left\{ \langle x, A_i u_i\rangle - \varphi_i(u_i) - \gamma b_{\mathcal{U}_i}(u_i) \right\} \right\}. \tag{30}$$

Let us denote by $u^*_{\gamma,i}(A_i^\top x)$ the unique solution of the subproblem $i$ in (30) for $i = 1, \cdots, m$. Then, under the decomposable structure, the evaluation of $f_\gamma$ and $u^*_\gamma(A^\top x) := [u^*_{\gamma,1}(A_1^\top x), \cdots, u^*_{\gamma,m}(A_m^\top x)]$ can be computed in parallel.

If we apply Algorithm 1 to solve (1) with the structure (29), then we have the following guarantee on the objective residual:

$$F(x^k) - F^\star \le \frac{L_A\|x^0 - x^\star\|^2}{2\gamma_1 k} + \frac{\gamma_1 D_{\mathcal{U}}}{k}\left(3 + (L_b - 1)\left(\ln(k+1) + 1\right)\right),$$

where $L_A := \sum_{i=1}^m \|A_i\|^2$, $L_b := \max\{L_{b_i} : 1 \le i \le m\}$ and $D_{\mathcal{U}} := \sum_{i=1}^m D_{\mathcal{U}_i}$. Hence, the convergence rate of Algorithm 1 stated in Theorem 1 is $\mathcal{O}\left(\frac{\ln(k)}{k}\right)$. If we choose $b_{\mathcal{U}_i}(\cdot) := (1/2)\|\cdot - \bar{u}_i^c\|^2$ for all $i = 1, \cdots, m$, then $L_b = 1$. Consequently, we obtain the $\mathcal{O}\left(\frac{L_A R_0\sqrt{D_{\mathcal{U}}}}{\varepsilon}\right)$-worst-case iteration-complexity.

### 4.4 The Lipschitz gradient structure

If $g$ is smooth and its gradient $\nabla g$ is Lipschitz continuous with the Lipschitz constant $L_g > 0$, then $F_\gamma := f_\gamma + g \in \mathcal{F}_L^{1,1}$, i.e., $\nabla F_\gamma$ is Lipschitz continuous with the Lipschitz constant $L_{F_\gamma} := L_g + \gamma^{-1}\|A\|^2$.

We replace the proximal-gradient step (13) using in Algorithm 1 by the following "full" gradient step

$$x^{k+1} := \hat{x}^k - \beta_{k+1}\big(\nabla g(\hat{x}^k) + Au^*_{\gamma_{k+1}}(A^\top\hat{x}^k)\big), \tag{31}$$

where $u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)$ is computed by (7) and $\beta_{k+1} := \frac{1}{L_g + \gamma_{k+1}^{-1}\|A\|^2}$ is a given step-size. Unlike (21), we update the parameters $\tau_k$ and $\gamma_k$ as

$$\tau_k := \frac{1}{k+1} \quad \text{and} \quad \gamma_{k+1} := \frac{k\gamma_k\|A\|^2}{L_g\gamma_k + \|A\|^2(k+1)},$$

where $\gamma_1 := \frac{\|A\|^2}{L_g}$ is fixed. We name this variant as Algorithm 1(b).

The following corollary summarizes the convergence properties of this variant, whose proof can be found in Appendix A.4.

**Corollary 1** *Assume that $g \in \mathcal{F}_L^{1,1}$ with the Lipschitz constant $L_g \ge 0$. Let $\{x^k\}$ be the sequence generated by Algorithm* 1(b)*. Then, for $k \ge 1$, one has*

$$F(x^k) - F^\star \le \frac{3L_g}{2k}\|x^0 - x^\star\|^2 + \frac{\|A\|^2}{L_g k}\left(\frac{2L_b}{L_g} + 1\right)D_{\mathcal{U}} + \frac{(L_b - 1)\|A\|^2}{L_g^2 k}\left(\ln(k) + 1\right)D_{\mathcal{U}}. \tag{32}$$

*If we choose $b_{\mathcal{U}}$ such that $L_b = 1$, then* (32) *reduces to*

$$F(x^k) - F^\star \le \frac{3L_g}{2k}\|x^0 - x^\star\|^2 + \frac{\|A\|^2}{L_g^2 k}(L_g + 2)D_{\mathcal{U}}.$$

*Consequently, the worst-case iteration-complexity of Algorithm* 1(b) *is $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$.*

## 5 Application to general constrained convex optimization

In this section, we customize Algorithm 1 to solve the following general constrained convex optimization problem:

$$\varphi^\star := \min_{u\in\mathbb{R}^n}\Big\{\varphi(u) : Au - b \in -\mathcal{K},\ u\in\mathcal{U}\Big\}, \tag{33}$$

where $\varphi$ is a proper, closed and convex function from $\mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$, $A\in\mathbb{R}^{p\times n}$, $b\in\mathbb{R}^p$, $\mathcal{U}$ and $\mathcal{K}$ are two nonempty, closed and convex set in $\mathbb{R}^n$ and $\mathbb{R}^p$, respectively. Without loss of generality, we can assume that $\varphi$ and $\mathcal{U}$ are decomposable as in (29) with $m\geq 1$.

Associated with the primal setting (33), we consider its dual problem

$$F^\star := \min_{x\in\mathbb{R}^p}\Big\{F(x) := \max_{u\in\mathcal{U}}\{\langle x, Au\rangle - \varphi(u)\} - \langle b, x\rangle + \max_{r\in\mathcal{K}}\langle x, r\rangle\Big\}. \tag{34}$$

Clearly, (34) has the same form as (1) with $f(x) := \max_u\{\langle x, Au\rangle - \varphi(u) : u\in\mathcal{U}\}$ and $g(x) := s_{\mathcal{K}}(x) - \langle b, x\rangle$, where $s_{\mathcal{K}}$ is the support function of $\mathcal{K}$.

We now specify Algorithm 1 to solve this dual problem. Computing $u^*_\gamma(x)$ requires to solve the following sub-problem:

$$u^*_\gamma(x) := \operatorname{argmin}_u\left\{\langle x, Au\rangle - \varphi(u) - \gamma b_{\mathcal{U}}(u)\right\}.$$

The proximal-step of $g$ becomes $\operatorname{prox}_g(x) := \operatorname{prox}_{s_{\mathcal{K}}}(x+b) = (x+b) - \operatorname{proj}_{\mathcal{K}}(x+b)$, where $\operatorname{proj}_{\mathcal{K}}(\cdot)$ is the projection onto $\mathcal{K}$. Together with the dual steps, we use an adaptive weighted averaging scheme

$$\bar{u}^k := \Gamma_k^{-1}\sum_{i=0}^k \tau_i^{-1}\gamma_{i+1}u^*_{\gamma_{i+1}}(\hat{x}^i), \quad \text{and} \quad \Gamma_k := \sum_{i=0}^k \tau_i^{-1}\gamma_{i+1}, \tag{35}$$

to construct an approximate primal solution $\bar{u}^k$ to an optimal solution $u^\star$ of (33). Clearly, we can compute $\bar{u}^k$ recursively starting from $\bar{u}^0 := \mathbf{0}^n$ as

$$\bar{u}^k := (1-\nu_k)\bar{u}^{k-1} + \nu_k u^*_{\gamma_{k+1}}(\hat{x}^k), \quad \text{where} \quad \nu_k := (\Gamma_k\tau_k)^{-1}\gamma_{k+1}\in(0,1]. \tag{36}$$

We incorporate this scheme into Algorithm 1 to solve (33). While Algorithm 1 constructs an approximate solution to the dual problem (34), (36) allows us to recover an approximate solution $\bar{u}^k$ of the primal problem (33). We name this algorithmic variant as Algorithm 1(c).

We specify the convergence guarantee of Algorithm 1(c) in the following theorem. The proof of this theorem is given in Appendix A.5.

**Theorem 2** *Assume that $b_{\mathcal{U}}$ is chosen such that $L_b = 1$, and $\bar{c} = 1$ in (21). Let $\{(x^k, \bar{u}^k)\}$ be generated by Algorithm 1(c). Then $\{\bar{u}^k\}\subset\mathcal{U}$ and*

$$\begin{cases} -\|x^\star\|\operatorname{dist}\left(b - A\bar{u}^k, \mathcal{K}\right) \leq \varphi(\bar{u}^k) - \varphi^\star \leq \frac{\|A\|^2\|x^0\|^2 + 2(\gamma_1+2\gamma_1^2)D_{\mathcal{U}}}{\gamma_1(k+1)}, \\ \operatorname{dist}\left(b - A\bar{u}^k, \mathcal{K}\right) \leq \frac{\|A\|^2\left(\|x^0 - x^\star\| + \sqrt{\|x^0-x^\star\|^2 + 2\|A\|^{-2}(2\gamma_1^2+\gamma_1)D_{\mathcal{U}}}\right)}{\gamma_1(k+1)}. \end{cases} \tag{37}$$

*Consequently, the worst-case iteration-complexity of Algorithm 1(c) to achieve an $\varepsilon$-solution $\bar{u}^k$ such that $|\varphi(\bar{u}^k) - \varphi^\star| \leq \varepsilon$ and $\operatorname{dist}\left(b - A\bar{u}^k, \mathcal{K}\right)\leq\varepsilon$ is $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$.*

Theorem 2 shows that Algorithm 1(c) has the $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ worst-case iteration-complexity on the primal objective residual and feasibility violation for (33).

## 6 Preliminarily numerical experiments

We demonstrate the performance of Algorithm 1 for solving the three well-known convex optimization problems. The first example is a LASSO problem with $\ell_1$-loss [34], the second one is a square-root LASSO studied in [8], and the last example is an image deblurring problem with a non-smooth data fidelity function (e.g., the $\ell_1$-norm or the $\ell_2$-norm function).

### 6.1 The $\ell_1$-$\ell_1$-regularized LASSO

We consider the $\ell_1$-$\ell_1$-regularized LASSO problem studied in [34] as follows:

$$F^\star := \min\left\{F(x) := \|Bx - b\|_1 + \lambda\|x\|_1 : x \in \mathbb{R}^p\right\}, \tag{38}$$

where $B$ and $b$ are defined as in (39), and $\lambda > 0$ is a regularization parameter.

The function $f(x) := \|Bx - b\|_1 = \max\left\{\langle B^\top u, x\rangle - \langle b, u\rangle : \|u\|_\infty \le 1\right\}$ falls into the decomposable case considered in Subsection 4.3. Hence, we can smooth $f$ using the quadratic prox-function to obtain

$$f_\gamma(x) := \max_u\left\{\langle x, B^\top u\rangle - \langle b, u\rangle - (\gamma/2)\|u\|^2 : u \in \mathcal{B}_\infty\right\}.$$

Clearly, we can show that $u^*_\gamma(Bx) := \mathrm{proj}_{\mathcal{B}_\infty}\left(\gamma^{-1}(Bx - b)\right)$. In this case, we also have $D_{\mathcal{B}_\infty} := \frac{1}{2}n$ and $\mathcal{U} := \mathcal{B}_\infty$.

Now, we apply Algorithm 1 to solve problem (39). To verify the theoretical bound in Theorem 1, we use CVX [18] to solve (39) and obtain a high accuracy approximate solution $x^\star$. Then, we can compute $R_0 := \|x^0 - x^\star\|_2$, and choose $\gamma_1 \equiv \gamma_1^* := \frac{\|B\|R_0}{\sqrt{6D_{\mathcal{B}_\infty}}}$. From Theorem 1, we have $F(x^k) - F^\star \le \frac{R_0\|B\|\sqrt{6D_{\mathcal{B}_\infty}}}{k}$, which is the worst-case bound of Algorithm 1, where $k$ is the iteration number.

For our comparison, we also implement the smoothing algorithm in [24] using the quadratic prox-function. As indicated in (5), we set $\gamma \equiv \gamma^* := \frac{\sqrt{2}\|B\|R_0}{\sqrt{D_{\mathcal{U}}}(k+1)}$. Hence, we also obtain the theoretical upper bound $F(x^k) - F^\star \le \frac{2\sqrt{2}\|B\|R_0\sqrt{D_{\mathcal{U}}}}{(k+1)}$. We name this algorithm as Non-adapt. Alg. (non-adaptive algorithm).

The test data is generated as follows: Matrix $B \in \mathbb{R}^{n\times p}$ is generated randomly using the standard Gaussian distribution $\mathcal{N}(0,1)$. We consider two cases. In the first case, we use non-correlated data, while in the second case, we generate $B$ with 50% correlated columns as $B(:, j+1) = 0.5B(:, j) + \mathrm{randn}(:)$. The observed measurement vector $b$ is generated as $b := Bx^\natural + \mathcal{N}(0, 0.05)$, where $x^\natural$ is a given $s$-sparse vector generated randomly using $\mathcal{N}(0,1)$.

We test both algorithms: Algorithm 1 and Non-adapt. Alg. on two problem instances of the size $(p, n, s) = (1000, 350, 100)$ (with and without correlated data, respectively). We sweep along the values of $\lambda$ to find an optimal value for $\lambda$ which are $\lambda = 6.2105$ for non-correlated data, and $\lambda = 5.7368$ for correlated data, respectively. For comparison, we first select the optimal value for $\gamma_1 := \gamma_1^*$ and $\gamma := \gamma^*$ in both algorithms. Then, we consider two cases: $(i)$ $\gamma_1 := 10\gamma_1^*$ and $\gamma := 10\gamma^*$, and $(ii)$ $\gamma_1 := 0.1\gamma_1^*$ and $\gamma := 0.1\gamma^*$.

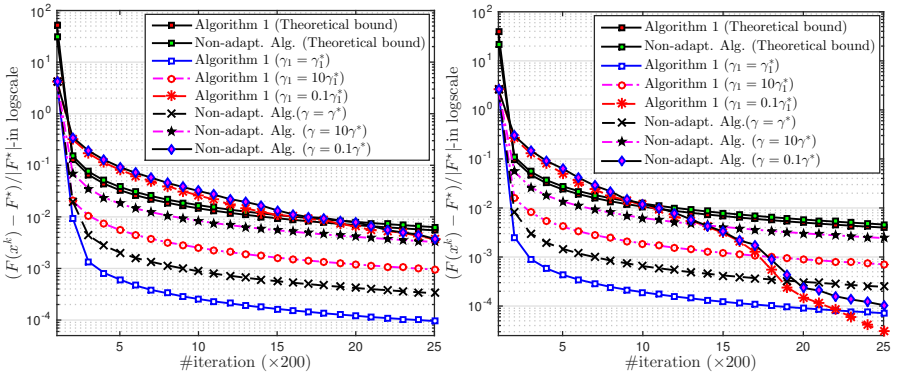


**Fig. 1** The empirical performance vs. the theoretical bounds of the 6 algorithmic variants (Left: non-correlated data, Right: correlated data).

Figure 1 plots the empirical bounds of the 6 variants vs. the theoretical bounds from 200 to 10,000 iterations. Obviously, both algorithms show their empirical rate which is much better than their theoretical bound. But if we change the smoothness parameters, the guarantee is no longer preserved. Algorithm 1 shows a better performance than Non-adapt. Alg. in both cases.

### 6.2 Square-root LASSO

We consider the following well-known square-root LASSO problem:

$$\min_{x\in\mathbb{R}^p}\Big\{F(x) := \|Bx - b\|_2 + \lambda\|x\|_1\Big\}. \tag{39}$$

As proved in [8], if matrix $B$ is Gaussian, then we can select the regularization parameter $\lambda$ such that we can obtain exact recovery to the true solution $x^\natural$.

The function $f$ defined by $f(x) := \|Bx - b\|_2$ can be written as

$$f(x) = \max_u\left\{\langle B^\top u, x\rangle - \langle b, u\rangle : \|u\|_2 \le 1\right\}.$$

Let $\mathcal{B}_2 := \{u \in \mathbb{R}^n : \|u\|_2 \le 1\}$ be the $\ell_2$-norm ball. We choose $b(u) := \frac{1}{2}\|u\|_2^2$ as a prox-function for $\mathcal{B}_2$. Then, we can smooth $f$ using $b(\cdot) := \frac{1}{2}\|\cdot\|_2^2$ as

$$f_\gamma(x) := \max_u\left\{\langle x, B^\top u\rangle - \langle b, u\rangle - (\gamma/2)\|u\|_2^2 : u \in \mathcal{B}_2\right\}.$$

Clearly, $u_\gamma^*(x) := \mathrm{proj}_{\mathcal{B}_2}\left(\gamma^{-1}(Bx - b)\right)$ is the solution of the maximization problem, where $\mathrm{proj}_{\mathcal{B}_2}$ is the projection onto $\mathcal{B}_2$. Moreover, we have $D_{\mathcal{U}} = \frac{1}{2}$.

Now, we apply Algorithm 1 to solve problem (39). We choose $\bar{c} := 1$ and set $\gamma_1 \equiv \gamma_1^* := \frac{\|A\|R_0}{\sqrt{6D_{\mathcal{U}}}}$, where $R_0 := \|x^0 - x^\star\|_2$. We also estimate the theoretical upper bound indicated in Theorem 1 for $F(x^k) - F^\star$ using (25), which is $\frac{\|A\|R_0\sqrt{6D_{\mathcal{U}}}}{k}$. We implement the smoothing algorithm in [24] for our comparison by using the same prox-function. The parameter of this algorithm is set as in the previous example.

The data test is generated as in Subsection 6.1. We also perform the test on two problem instances of size $(p, n, s) = (1000, 350, 100)$: non-correlated data and correlated data. We choose the regularization parameter $\lambda$ as suggested in [8]. We use the same setting for the smoothness parameter $\gamma$ in both algorithms

as in Subsection 6.1. In this case, the theoretical upper bound of Algorithm 1 depends on the log-term which is scaled by the condition number of $BB^\top$, and is worse than in Non-adapt. Alg. variant.

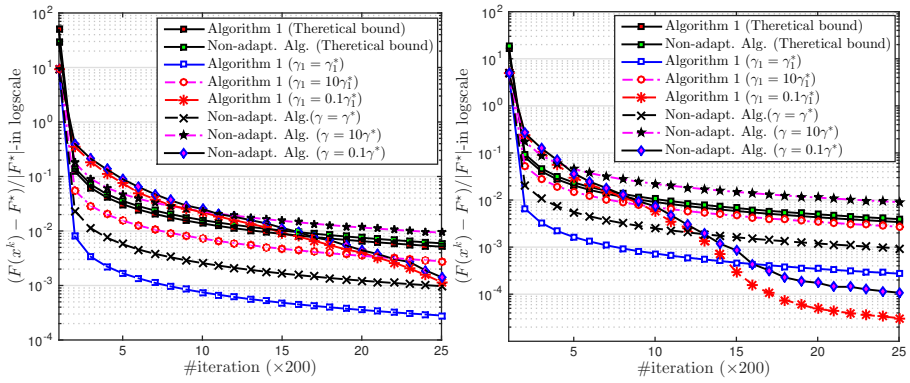


**Fig. 2** The empirical performance vs. the theoretical bounds of the 6 algorithmic variants (Left: non-correlated data, Right: 50%-correlated collumns).

Figure 2 plots the empirical bounds of the 6 variants vs. the theoretical bounds from 200 to 10,000 iterations. Obviously, both algorithms show their empirical rate which is much better than their theoretical bound. Algorithm 1 gives a better performance than the nonadaptive method in this example. We note that the theoretical bound in Algorithm 1 remains non-optimal, while it is optimal in the nonadaptive one.

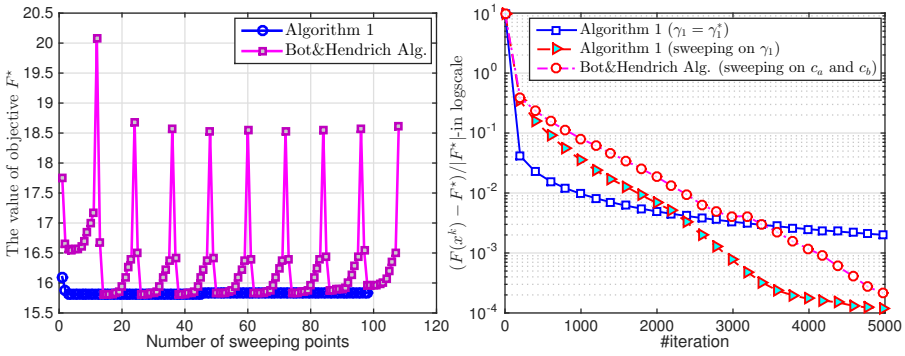


**Fig. 3** Comparison of Algorithm 1 and `Boţ&Hendrich Alg.` (Left: the objective values vs. the number of sweeping points, Right: Convergence of the relative objective residual).

Finally, we compare Algorithm 1 with the variable smoothing algorithm in [11] (`Boţ&Hendrich Alg.`). Whlile the first term $f(x) := \|Ax - b\|_2$ is smoothened as in Algorithm 1, we smooth the second term $g(x) := \lambda\|x\|_1$ as

$$g_\beta(x) := \max_v \left\{\langle x, v\rangle - (\beta/2)\|v\|_2^2 : \|v\|_\infty \le 1\right\}.$$

Then, we update $\gamma_k$ and $\beta_k$ as $\gamma_k = \frac{1}{c_a(k+1)}$ and $\beta_k = \frac{1}{c_b(k+1)}$, respectively as suggested in [11], where $c_a$ and $c_b$ are two appropriate constants.

We compare Algorithm 1 and `Boţ&Hendrich Alg.` on a problem instance of size $(p, n, s) = (1000, 350, 100)$, where the data is generated as in the previous tests. To find an appropriate value of $c_a$ and $c_b$, we sweep $c_a \in [10, 5000]$ simultaneously with $c_b \in [0.001, 500]$. We obtain $c_a = 51$ and $c_b = 49$. For Algorithm 1, we consider two cases. In the first case, we set $\gamma_1 = \gamma_1^* = 129.5505$ computed from the worst-case bound, while in the second case, we also sweep $\gamma_1 \in [10, 1000]$ to find an appropriate value $\gamma_1 = 51$. The results of both algorithms are plotted in Figure 3 for 5000 iterations.

Figure 3 (left) shows that the objective value produced by Algorithm 1 does not vary much when $\gamma_1 \in [10, 1000]$, while, in `Boţ&Hendrich Alg.`, the objective value changes rapidly when we sweep on $c_a$ and $c_b$ simultaneously. Hence, it is unclear how to choose an appropriate value for $c_a$ and $c_b$ without sweeping. Figure 3 (right) shows the convergence behavior of both algorithms. Without sweeping, Algorithm 1 has a good empirical convergence rate in the early iterations. With sweeping, both algorithms perform better in the later iterations. Algorithm 1 has a better performance than `Boţ&Hendrich Alg.`.

6.3 Image deblurring with the $\ell_1$ or $\ell_2$-data fidelity function

We consider an image deblurring problem using the $\ell_\alpha$-norm fidelity term as

$$\min_X \left\{ F(X) := \|\mathcal{A}(X) - b\|_\alpha + \lambda \|WX\|_1 : X \in \mathbb{R}^{m\times q} \right\}, \tag{40}$$

where $\alpha \in \{1, 2\}$, $A : \mathbb{R}^p \to \mathbb{R}^p$ $(p = m \times q)$ is a blurring kernel, $b$ is an observed noisy image, $W : \mathbb{R}^p \to \mathbb{R}^p$ is the orthogonal Haar wavelet transform with four levels, $\lambda > 0$ is the regularizer parameter.

We now apply Algorithm 1 (`Alg. 1`) to solve problem (40) and compare it with the nonadaptive variant (`Nes. Alg.`) and Boţ & Hendrich's algorithm (`BH Alg.`) in [11]. Since $A$ is orthogonal, we can use the quadratic smoothing function as $b_{\mathcal{U}}(X) := (1/2)\|X\|_F^2$. With this choice, we can compute the gradient of $u_\gamma^*(X)$ defined by (7) as $u_\gamma^*(X) = \mathrm{proj}_{\mathcal{B}_\alpha^*}(\gamma^{-1}(\mathcal{A}(X) - b))$, where $\mathrm{proj}_{\mathcal{B}_\alpha}$ is the projection onto the dual norm ball $\mathcal{B}_\alpha^*$ of the $\ell_\alpha$-norm.

We test three algorithms on the five images: `cameraman`, `barbara`, `lena`, `boat` and `house` widely used in the literature. The noisy images are generated as in [4]. Although we use the non-smooth $\ell_\alpha$-norm function with $\alpha = 1$ or $\alpha = 2$, the regularization parameter $\lambda$ is set to $\lambda := 10^{-4}$ as suggested in [4], but it still provides the best recovery compared to other values in all 5 images.

While we fix $\gamma_1 = 62$ in Algorithm 1 which is roughly computed from the worst-case bound, we sweep $\gamma$ and $c_a$ (see Subsection 6.2) in $[0.0001, 1000]$ to choose the best possible value for `Nes. Alg.` and `BH Alg.` in each image (with 300 iterations). We also set $c_b = c_a$ as suggested in [11]. For `Nes. Alg.`, we have $\gamma = 1$ in the `boat` image, while in the other 4 images, $\gamma = 2.5$ is the best value. For `BH Alg.`, we have $c_a = 0.005$ in the `cameraman`, `barbara` and `boat` images, and $c_a = 0.0025$ in the `lena` and `house` images. The PSNR (Peak Signal to Noise Ratio [4]) of the 8 algorithms are reported in Table 1.

It shows that the nonsmooth $\ell_1$-norm objective produces slightly better recovery images in terms of PSNR than the $\ell_2$-norm objective in many cases for Algorithm 1, but it is not the case in `Nes. Alg.` and `BH Alg.` In addition,

**Table 1** The PSNR values reported by the 8 algorithmic variants on the 5 test images

| Images | cameraman | barbara | lena | boat | house |
|---|---|---|---|---|---|
| PSNR of 4 algorithms after 300 iterations | | | | | |
| Alg. 1 ($\ell_1$, $\gamma_1 = 62$) | 26.2140 | 26.8253 | 27.1793 | 26.4951 | 30.9848 |
| Alg. 1 ($\ell_1$, $\gamma_1$-sweeping) | 26.2693 | 27.0682 | 27.5440 | 26.5519 | 31.6877 |
| Alg. 1 ($\ell_2$, $\gamma_1 = 62$) | 26.2128 | 26.8232 | 27.1782 | 26.4923 | 30.4126 |
| Alg. 1 ($\ell_2$, $\gamma_1$-sweeping) | 26.2128 | 26.8232 | 27.1782 | 26.4923 | 30.4126 |
| Nes. Alg. ($\ell_1$, $\gamma$-sweeping) | 25.0601 | 26.1376 | 26.3776 | 25.2301 | 30.2982 |
| Nes. Alg. ($\ell_2$, $\gamma$-sweeping) | 25.0908 | 26.1361 | 26.3901 | 25.2364 | 30.4081 |
| BH Alg. ($\ell_1$, $c_a$-sweeping) | 25.5784 | 26.3421 | 26.5916 | 25.6025 | 31.1606 |
| BH Alg. ($\ell_2$, $c_a$-sweeping) | 25.4784 | 26.4421 | 26.5916 | 25.6025 | 31.1606 |
| PSNR of 4 algorithms after 500 iterations | | | | | |
| Alg. 1 ($\ell_1$, $\gamma_1 = 62$) | 27.0371 | 27.6286 | 28.1471 | 27.3116 | 32.1771 |
| Alg. 1 ($\ell_1$, $\gamma_1$-sweeping) | 27.1666 | 27.8449 | 28.2086 | 27.4410 | 32.8647 |
| Alg. 1 ($\ell_2$, $\gamma_1 = 62$) | 27.0363 | 27.6279 | 28.1486 | 27.3111 | 32.1710 |
| Alg. 1 ($\ell_2$, $\gamma_1$-sweeping) | 27.0363 | 27.6279 | 28.1486 | 27.3111 | 32.1710 |
| Nes. Alg. ($\ell_1$, $\gamma$-sweeping) | 25.0857 | 26.1686 | 26.4590 | 26.1321 | 30.4720 |
| Nes. Alg. ($\ell_2$, $\gamma$-sweeping) | 25.0845 | 26.1686 | 26.4582 | 25.2265 | 30.4718 |
| BH Alg. ($\ell_1$, $c_a$-sweeping) | 26.5030 | 27.1588 | 27.1630 | 27.0277 | 31.8824 |
| BH Alg. ($\ell_2$, $c_a$-sweeping) | 26.5030 | 27.1588 | 27.1630 | 27.0277 | 31.8824 |
| PSNR of 4 algorithms after 1000 iterations | | | | | |
| Alg. 1 ($\ell_1$, $\gamma_1 = 62$) | 27.4774 | 27.8353 | 28.4224 | 27.6596 | 32.9985 |
| Alg. 1 ($\ell_1$, $\gamma_1$-sweeping) | 27.3291 | 27.8659 | 28.4040 | 27.9482 | 33.2038 |
| Alg. 1 ($\ell_2$, $\gamma_1 = 62$) | 27.2524 | 27.8070 | 28.4774 | 27.5268 | 33.1879 |
| Alg. 1 ($\ell_2$, $\gamma_1$-sweeping) | 27.2524 | 27.8070 | 28.4774 | 27.5268 | 33.1879 |
| Nes. Alg. ($\ell_1$, $\gamma$-sweeping) | 25.0870 | 26.1691 | 26.4602 | 26.1371 | 30.4698 |
| Nes. Alg. ($\ell_2$, $\gamma$-sweeping) | 25.0867 | 26.1690 | 26.4600 | 25.2267 | 30.4700 |
| BH Alg. ($\ell_1$, $c_a$-sweeping) | 27.1128 | 27.8391 | 27.9327 | 27.3487 | 32.6715 |
| BH Alg. ($\ell_2$, $c_a$-sweeping) | 27.1723 | 27.8205 | 27.9327 | 27.3143 | 32.6715 |

Algorithm 1 is superior to `Nes. Alg.` in all cases, and is also better than `BH Alg.` in the majority of the test. We note that the complexity-per-iteration of the four algorithms are essentially the same, while our new adaptive strategy produces better solutions in terms of PSNR than the other two methods. In addition, our algorithm significantly improves the PSNR if we run it further, while the nonadaptive variant does not make any clear progress on the PSNR value if we continue running it. If we sweep the values of $\gamma_1$ in Algorithm 1 ($\gamma_1$-sweeping), we can also improve the results of this algorithm.

**Acknowledgements** This research was supported by NSF, Grant No. IPF 16-4829.

## A Appendix: The proof of technical results

This appendix provides the full proof of the technical results presented in the main text.

### A.1 The proof of Lemma 2: Descent property of the proximal gradient step

By using (9) with $f_\gamma(x) := \varphi^*_\gamma(A^\top x)$, $\nabla f_\gamma(\bar{x}) = A\nabla\varphi^*_\gamma(A^\top\bar{x})$, $z := A^\top x$, $\bar{z} := A^\top\bar{x}$, and $\|A^\top(x-\bar{x})\| \le \|A\|\|x-\bar{x}\|$ we can show that

$$\frac{\gamma}{2}\|u^*_\gamma(A^\top x) - u^*_\gamma(A^\top\bar{x})\|^2 \le f_\gamma(x) - f_\gamma(\bar{x}) - \langle\nabla f_\gamma(x), x-\bar{x}\rangle \le \frac{\|A\|^2}{2\gamma}\|x-\hat{x}\|^2.$$

Using this estimate, we can show that the proof of (14) can be done similarly as in [31]. □

## A.2 The proof of Lemma 3: Key estimate

We first substitute $\beta = \frac{\gamma_{k+1}}{\|A\|^2}$ into (14) and using (15) to obtain

$$\begin{aligned} F_{\gamma_{k+1}}(x^{k+1}) \le & F_{\gamma_{k+1}}(x^k) - \tfrac{\gamma_{k+1}}{2}\|u^*_{\gamma_{k+1}}(A^\top x^k) - u^*_{\gamma_{k+1}}(A^\top \hat{x}^k)\|^2 \\ & + \tfrac{\|A\|^2}{\gamma_{k+1}}\left\langle x^{k+1} - \hat{x}^k, x - \bar{x}^k\right\rangle - \tfrac{\|A\|^2}{2\gamma_{k+1}}\|\hat{x}^k - x^{k+1}\|^2. \end{aligned}$$

Multiplying this inequality by $(1-\tau_k)$ and (14) by $\tau_k$, and summing up the results we obtain

$$\begin{aligned} F_{\gamma_{k+1}}(x^{k+1}) \le & (1-\tau_k)F_{\gamma_{k+1}}(x^k) + \tau_k\hat{\ell}^k_{\gamma_{k+1}}(x) - \tfrac{(1-\tau_k)\gamma_{k+1}}{2}\|u^*_{\gamma_{k+1}}(A^\top x^k) - u^*_{\gamma_{k+1}}(A^\top \hat{x}^k)\|^2 \\ & + \tfrac{\|A\|^2}{\gamma_{k+1}}\left\langle \hat{x}^k - x^{k+1}, \hat{x}^k - (1-\tau_k)x^k - \tau_k x\right\rangle - \tfrac{\|A\|^2}{2\gamma_{k+1}}\|\hat{x}^k - x^{k+1}\|^2, \end{aligned}$$

where $\hat{\ell}^k_\gamma(x) := f_\gamma(\hat{x}^k) + \langle \nabla f_\gamma(\hat{x}^k), x - \hat{x}^k\rangle + g(x)$.

From (16), we have $\tau_k\tilde{x}^k = \hat{x}^k - (1-\tau_k)x^k$, we can write this inequality as

$$\begin{aligned} F_{\gamma_{k+1}}(x^{k+1}) \le & (1-\tau_k)F_{\gamma_{k+1}}(x^k) + \tau_k\hat{\ell}^k_{\gamma_{k+1}}(x) - \frac{(1-\tau_k)\gamma_{k+1}}{2}\|u^*_{\gamma_{k+1}}(A^\top x^k) - u^*_{\gamma_{k+1}}(A^\top \hat{x}^k)\|^2 \\ & + \frac{\|A\|^2\tau_k^2}{2\gamma_{k+1}}\Big[\|\tilde{x}^k - x\|^2 - \|\tilde{x}^k - \tau_k^{-1}(\hat{x}^k - x^{k+1}) - x\|^2\Big]. \end{aligned} \tag{41}$$

Using (10) with $\gamma := \gamma_{k+1}$, $\hat{\gamma} := \gamma_k$ and $z := A^\top x^k$, we get

$$f_{\gamma_{k+1}}(x^k) \le f_{\gamma_k}(x^k) + (\gamma_k - \gamma_{k+1})b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top x^k)),$$

which leads to ($cf$: $F_\gamma = f_\gamma + g$):

$$F_{\gamma_{k+1}}(x^k) \le F_{\gamma_k}(x^k) + (\gamma_k - \gamma_{k+1})b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top x^k)). \tag{42}$$

Next, we estimate $\hat{\ell}^k_{\gamma_{k+1}}$. Using the definition of $f_\gamma$ and $\nabla f_\gamma$, we can deduce

$$\begin{aligned} \hat{\ell}^k_{\gamma_{k+1}}(x) &:= f_{\gamma_{k+1}}(\hat{x}^k) + \left\langle \nabla f_{\gamma_{k+1}}(\hat{x}^k), x - \hat{x}^k\right\rangle + g(x) \\ &= \left\langle \hat{x}^k, A^\top u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)\right\rangle - \varphi(u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)) - \gamma_{k+1}b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)) \\ &\quad + \left\langle x - \hat{x}^k, Au^*_{\gamma_{k+1}}(A^\top\hat{x}^k)\right\rangle + g(x) \\ &= \left\langle x, Au^*_{\gamma_{k+1}}(A^\top\hat{x}^k)\right\rangle - \varphi(u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)) - \gamma_{k+1}b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)) + g(x) \\ &\le \max_u\left\{\left\langle x, Au\right\rangle - \varphi(u) : u \in \mathcal{U}\right\} - \gamma_{k+1}b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)) + g(x) \\ &= F(x) - \gamma_{k+1}b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)). \end{aligned} \tag{43}$$

Substituting $\tilde{x}^{k+1} := \tilde{x}^k - \frac{1}{\tau_k}(\hat{x}^k - x^{k+1})$ from the third line of (16) together with (42), and (43) into (41), we can derive

$$F_{\gamma_{k+1}}(x^{k+1}) \le (1-\tau_k)F_{\gamma_k}(x^k) + \tau_k F(x) + \tfrac{\|A\|^2\tau_k^2}{2\gamma_{k+1}}\left[\|\tilde{x}^k - x\|^2 - \|\tilde{x}^{k+1} - x\|^2\right] - R_k,$$

which is indeed (18), where $R_k$ is given by (19).

Finally, we prove (20). Indeed, using the strong convexity and the $L_b$-smoothness of $b_{\mathcal{U}}$, we can lower bound

$$\begin{aligned} R_k \ge & \tfrac{\tau_k\gamma_{k+1}}{2}\|u^*_{\gamma_{k+1}}(A^\top\hat{x}^k) - \bar{u}^c\|^2 + \tfrac{(1-\tau_k)\gamma_{k+1}}{2}\|u^*_{\gamma_{k+1}}(A^\top x^k) - u^*_{\gamma_{k+1}}(A^\top\hat{x}^k)\|^2 \\ & - \tfrac{L_b}{2}(1-\tau_k)(\gamma_k - \gamma_{k+1})\|u^*_{\gamma_{k+1}}(A^\top x^k) - \bar{u}^c\|^2. \end{aligned}$$

Letting $\hat{v}_k := u^*_{\gamma_{k+1}}(A^\top\hat{x}^k) - \bar{u}^c$ and $v_k := u^*_{\gamma_{k+1}}(A^\top x^k) - \bar{u}^c$, we write $R_k$ as

$$\begin{aligned}2\gamma_{k+1}^{-1}R_k &\geq \tau_k\|\hat{v}_k\|^2 + (1-\tau_k)\|\hat{v}_k - v_k\|^2 - (1-\tau_k)(\gamma_{k+1}^{-1}\gamma_k - 1)L_b\|v_k\|^2\\ &= \|\hat{v}_k\|^2 - 2(1-\tau_k)\langle\hat{v}_k, v_k\rangle + (1-\tau_k)\big[1 - (\gamma_{k+1}^{-1}\gamma_k - 1)L_b\big]\|v_k\|^2\\ &= \|\hat{v}_k - (1-\tau_k)v_k\|^2 + (1-\tau_k)\left[\tau_k - (\gamma_{k+1}^{-1}\gamma_k - 1)L_b\right]\|v_k\|^2\\ &\geq (1-\tau_k)\left[\tau_k - (\gamma_{k+1}^{-1}\gamma_k - 1)L_b\right]\|v_k\|^2,\end{aligned}$$

which obviously implies (20). □

## A.3 The proof of Lemma 4: The choice of parameters

First, using the update rules of $\tau_k$ and $\gamma_k$ in (21), we can express the quantity $m_k$ as

$$m_k := \frac{\gamma_{k+1}(1-\tau_k)\left[\gamma_{k+1}\tau_k - L_b(\gamma_k - \gamma_{k+1})\right]}{\tau_k^2} = -\frac{\gamma_1^2\bar{c}^2\left[(L_b-1)(k+\bar{c})+1\right]}{(k+\bar{c})^2}.$$

Moreover, it follows from the properties of $b_{\mathcal{U}}$ that

$$\frac{1}{2}\|u^*_{\gamma_{k+1}}(A^\top x^k) - \bar{u}^c\|^2 \leq b_{\mathcal{U}}(u^*_{\gamma_{k+1}}(A^\top x^k)) \leq D_{\mathcal{U}}.$$

Multiplying the lower bound (20) by $\frac{\gamma_{k+1}}{\tau_k^2}$ and combining the result with the last inequality and the estimate of $m_k$, we obtain the first lower bound in (22).

Next, using the update rules (21) of $\tau_k$ and $\gamma_k$, we have $\frac{(1-\tau_k)\gamma_{k+1}}{\tau_k^2} = \frac{(k+\bar{c}-1)(k+\bar{c})^2\gamma_1\bar{c}}{(k+\bar{c})(k+\bar{c})} = \frac{\gamma_1\bar{c}(k+\bar{c}-1)^2}{(k+\bar{c}-1)} = \frac{\gamma_k}{\tau_{k-1}^2}$, which is the second equality in (22).

Using (18) with the lower bound of $R_k$ from (22), we have

$$\frac{\gamma_{k+1}}{\tau_k^2}\Delta F_{k+1} + \frac{\|A\|^2}{2}\|\tilde{x}^{k+1} - x^\star\|^2 \leq \frac{(1-\tau_k)\gamma_{k+1}}{\tau_k^2}\Delta F_k + \frac{\|A\|^2}{2}\|\tilde{x}^k - x^\star\|^2 + s_k D_{\mathcal{U}}, \tag{44}$$

where $\Delta F_k := F_{\gamma_k}(x^k) - F^\star$ and $s_k := \frac{\gamma_1^2\bar{c}^2[(L_b-1)(k+\bar{c})+1]}{(k+\bar{c})^2}$. Using this inequality and the relation $\frac{(1-\tau_k)\gamma_{k+1}}{\tau_k^2} = \frac{\gamma_k}{\tau_{k-1}^2}$ in (22), we can easily show that

$$\frac{\gamma_{k+1}}{\tau_k^2}\Delta F_{k+1} + \frac{\|A\|^2}{2}\|\tilde{x}^{k+1} - x^\star\|^2 \leq \frac{\gamma_k}{\tau_{k-1}^2}\Delta F_k + \frac{\|A\|^2}{2}\|\tilde{x}^k - x^\star\|^2 + s_k D_{\mathcal{U}}.$$

By induction, we obtain from the last inequality that

$$\frac{\gamma_{k+1}}{\tau_k^2}\Delta F_{k+1} + \frac{\|A\|^2}{2}\|\tilde{x}^{k+1} - x^\star\|^2 \leq \frac{(1-\tau_0)\gamma_1}{\tau_0^2}\Delta F_0 + \frac{\|A\|^2}{2}\|\tilde{x}^0 - x^\star\|^2 + S_k D_{\mathcal{U}}, \tag{45}$$

which implies (23), where $S_k := \sum_{i=0}^k s_k = \gamma_1^2\bar{c}^2\sum_{i=0}^k \frac{[(L_b-1)(i+\bar{c})+1]}{(i+\bar{c})^2}$.

Finally, to prove (24), we use two elementary inequalities $\sum_{i=1}^{k+\bar{c}}\frac{1}{i} < 1 + \ln(k+\bar{c})$ and $\sum_{i=0}^k \frac{1}{(i+\bar{c})^2} \leq \frac{1}{\bar{c}^2} + \sum_{i=1}^k \frac{1}{(i+\bar{c}-1)(i+\bar{c})} < \frac{1}{\bar{c}^2} + \frac{1}{\bar{c}}$. □

## A.4 The proof of Corollary 1: The smooth accelerated gradient method

First, it is similar to the proof of (44), we can derive

$$\frac{\gamma_{k+1}}{(L_g\gamma_{k+1} + \|A\|^2)\tau_k^2}\Delta F_{k+1} + \frac{1}{2}\|\tilde{x}^{k+1} - x^\star\|^2 \leq \frac{(1-\tau_k)\gamma_{k+1}}{(L_g\gamma_{k+1} + \|A\|^2)\tau_k^2}\Delta F_k + \frac{1}{2}\|\tilde{x}^k - x^\star\|^2 + \hat{s}_k D_{\mathcal{U}},$$

where $\Delta F_k := F_{\gamma_k}(x^k) - F^\star$, and $\hat{s}_k := \frac{\gamma_{k+1}(1-\tau_k)\left[L_b(\gamma_k - \gamma_{k+1}) - \gamma_{k+1}\tau_k\right]}{\tau_k^2(L_g\gamma_{k+1} + \|A\|^2)}$.

Next, we impose condition $\frac{(1-\tau_k)\gamma_{k+1}}{\tau_k^2(L_g\gamma_{k+1}+\|A\|^2)} = \frac{\gamma_k}{\tau_{k-1}^2(L_g\gamma_k+\|A\|^2)}$ and choose $\tau_k = \frac{1}{k+1}$. Then, we can show from the last condition that $\gamma_{k+1} = \frac{k\gamma_k\|A\|^2}{L_g\gamma_k+\|A\|^2(k+1)}$. Now, we show that $\gamma_k \le \frac{\gamma_1}{k+1}$. Indeed, we have $\frac{1}{\gamma_{k+1}} = \left(\frac{k+1}{k}\right)\frac{1}{\gamma_k} + \frac{L_g}{\|A\|^2 k} \ge \left(\frac{k+1}{k}\right)\frac{1}{\gamma_k}$, which implies that $\gamma_{k+1} \le \frac{k}{k+1}\gamma_k$. By induction, we get $\gamma_{k+1} \le \frac{\gamma_1}{k+1}$. On the other hand, assume that $\frac{1}{\gamma_{k+1}} = \left(\frac{k+1}{k}\right)\frac{1}{\gamma_k} + \frac{L_g}{\|A\|^2 k} \le \frac{1}{\gamma_k}\left(\frac{k}{k-1}\right)$ for $k \ge 2$. This condition leads to $\gamma_k \le \frac{\|A\|^2}{L_g(k-1)}$. Using $\gamma_k \le \frac{\gamma_1}{k} = \frac{\|A\|^2}{L_g k}$ due to the choice of $\gamma_1$, we can show that $\gamma_k \le \frac{\|A\|^2}{L_g(k-1)}$. Hence, with the choice $\gamma_1 := \frac{\|A\|^2}{L_g}$, the estimate $\frac{1}{\gamma_{k+1}} \le \frac{1}{\gamma_k}\left(\frac{k}{k-1}\right)$ and the update rule of $\gamma_k$ eventually imply

$$\frac{\gamma_1\|A\|^2}{(L_g\gamma_1+2\|A\|^2)k} = \frac{\gamma_2}{k} \le \gamma_{k+1} \le \frac{\gamma_1}{k+1}, \quad \forall k \ge 1.$$

This condition leads to $\frac{\tau_{k-1}^2(L_g\gamma_k+\|A\|^2)}{\gamma_k} = L_g\tau_{k-1}^2 + \frac{\tau_{k-1}^2}{\gamma_k}\|A\|^2 \le \frac{L_g}{k^2} + \frac{3L_g(k-1)}{k^2} = \frac{3L_g}{k}$. Using the estimates of $\tau_k$ and $\gamma_k$, we can easily show that $\hat{s}_k \le \frac{L_b\|A\|^2}{L_g^2 k(k+2)} + \frac{L_b\|A\|^2}{L_g^2(k+1)(k+2)} + \frac{(L_b-1)\|A\|^2 k}{L_g(k+1)(k+2)}$. Hence, we can show that

$$\hat{S}_k := \sum_{i=0}^{k}\hat{s}_i \le \frac{2L_b\|A\|^2}{L_g^2} + \frac{(L_b-1)\|A\|^2}{L_g^2}\sum_{i=0}^{k}\frac{1}{i+1} = \frac{2L_b\|A\|^2}{L_g^2} + \frac{(L_b-1)\|A\|^2}{L_g^2}\left(\ln(k)+1\right).$$

Using this estimate, we can show that

$$F_{\gamma_k}(x^k) - F^\star \le \frac{3L_g}{2k}\|x^0-x^\star\|^2 + \frac{2L_b\|A\|^2}{L_g^2 k}D_{\mathcal{U}} + \frac{(L_b-1)\|A\|^2}{L_g^2 k}\left(\ln(k)+1\right)D_{\mathcal{U}}.$$

Finally, using the bound (11) and $\gamma_k \le \frac{\|A\|^2}{L_g(k+1)} < \frac{\|A\|^2}{L_g k}$, we obtain (32). □

## A.5 The proof of Theorem 2: Primal solution recovery

Let $\Delta F_k := F_{\gamma_k}(x^k) - F^\star$. Then, by (11), we have $\Delta F_k \ge F(x^k) - F^\star - \gamma_k D_{\mathcal{U}} \ge -\gamma_k D_{\mathcal{U}}$. Similar to the proof of Lemma 4, we can prove that

$$\frac{\gamma_{i+1}}{\tau_i^2}\Delta F_{i+1} \le \frac{\gamma_i}{\tau_{i-1}^2}\Delta F_i + \frac{\gamma_{i+1}}{\tau_i}\Delta\hat{\ell}_{\gamma_{i+1}}(x) + \frac{\|A\|^2}{2}\left(\|\tilde{x}^i-x\|^2 - \|\tilde{x}^{i+1}-x\|^2\right) + s_i D_{\mathcal{U}}, \quad (46)$$

where $s_i := \frac{[(L_b-1)(i+\bar{c})+1]}{(i+\bar{c})^2}$ as in the proof of Lemma 4, and $\Delta\hat{\ell}^k_{\gamma_{k+1}}(x) = \langle x, Au^*_{\gamma_{k+1}}(\hat{x}^k)\rangle - \varphi(u^*_{\gamma_{k+1}}(\hat{x}^k)) + g(x) - F^\star = \langle x, Au^*_{\gamma_{k+1}}(\hat{x}^k) - b\rangle - \varphi(u^*_{\gamma_{k+1}}(\hat{x}^k)) + s_{\mathcal{K}}(x) - F^\star$. Summing up this inequality from $i = 1$ to $i = k$ and using $\tau_0 = 1$ and $\tilde{x}^0 = x^0$, we obtain

$$\frac{\gamma_{k+1}}{\tau_k^2}\Delta F_{k+1} \le \sum_{i=1}^{k}\frac{\gamma_{i+1}}{\tau_i}\Delta\hat{\ell}_{\gamma_{i+1}}(x) + \frac{\|A\|^2}{2}\left(\|\tilde{x}^1-x\|^2 - \|\tilde{x}^{k+1}-x\|^2\right) + \gamma_1\Delta F_1 + S_k D_{\mathcal{U}}, \quad (47)$$

where $S_k := \sum_{i=1}^{k} s_i$. Now, using again (46) with $k = 1$, $x^0 = \tilde{x}^0$ and $\tau_0 = 1$, we get $\gamma_1\Delta F_1 \le \gamma_1\tau_0\Delta\hat{\ell}_{\gamma_1}(x) + \frac{\|A\|^2}{2}\left(\|x^0-x^\star\|^2 - \|\tilde{x}^1-x^\star\|^2\right)$. Using this into (47), one yields

$$\begin{aligned}\frac{\gamma_{k+1}}{\tau_k^2}\Delta F_{k+1} \le{}& \sum_{i=0}^{k}\frac{\gamma_{i+1}}{\tau_i}\left(\langle x, Au^*_{\gamma_{i+1}}(\hat{x}^i) - b\rangle - \varphi(u^*_{\gamma_{i+1}}(\hat{x}^i)) + s_{\mathcal{K}}(x) - F^\star\right)\\ &+ \frac{\|A\|^2}{2}\|x^0-x\|^2 + S_k D_{\mathcal{U}}. \end{aligned} \quad (48)$$

Combining $\bar{u}^k$ defined by (35) with $w_i := \frac{\gamma_{i+1}}{\tau_i}$, and the convexity of $\varphi$, we have

$$\sum_{i=0}^{k}\frac{\gamma_{i+1}}{\tau_i}\Big(\langle x, Au^*_{\gamma_{i+1}}(\hat{x}^i) - b\rangle - \varphi(u^*_{\gamma_{i+1}}(\hat{x}^i))\Big) \le \Gamma_k\Big(\langle x, A\bar{u}^k - b\rangle - \varphi(\bar{u}^k)\Big).$$

Substituting this into (48) and then using $\Delta F_{k+1} \ge -\gamma_{k+1}D_{\mathcal{U}}$ we get

$$-\frac{\gamma^2_{k+1}}{\tau^2_k}D_{\mathcal{U}} \le \Gamma_k\Big(\langle x, A\bar{u}^k - b\rangle - \varphi(\bar{u}^k) + s_{\mathcal{K}}(x) - F^\star\Big) + \frac{\|A\|^2}{2}\|x^0 - x\|^2 + S_kD_{\mathcal{U}},$$

which implies

$$F^\star \le \langle x, A\bar{u}^k - b\rangle - \varphi(\bar{u}^k) + s_{\mathcal{K}}(x) + \frac{\|A\|^2}{2\Gamma_k}\|x^0 - x\|^2 + \frac{D_{\mathcal{U}}}{\Gamma_k}\left(\frac{\gamma^2_{k+1}}{\tau^2_k} + S_k\right).$$

By arranging this inequality, we get

$$\inf_{r\in\mathcal{K}}\langle x, b - A\bar{u}^k - r\rangle + \varphi(\bar{u}^k) \le -F^\star + \frac{\|A\|^2}{2\Gamma_k}\|x^0 - x\|^2 + \frac{D_{\mathcal{U}}}{\Gamma_k}\left(\frac{\gamma^2_{k+1}}{\tau^2_k} + S_k\right), \tag{49}$$

where we use the relation $-s_{\mathcal{K}}(x) = -\sup_{r\in\mathcal{K}}\langle x, r\rangle = \inf_{r\in\mathcal{K}}\langle x, -r\rangle$. On the other hand, by the saddle point theory for the primal and dual problems (33) and (34), for any optimal solution $x^\star$, we can show that

$$-F^\star = \varphi^\star \le \varphi(u) - \langle x^\star, Au - b + r\rangle,\ \forall u \in \mathcal{U},\ r \in \mathcal{K}.$$

Since this inequality holds for any $r \in \mathcal{K}$ and $u \in \mathcal{U}$, by using $u = \bar{u}^k$, it leads to

$$\inf_{r\in\mathcal{K}}\langle x^\star, A\bar{u}^k - b + r\rangle - \varphi(\bar{u}^k) \le F^\star. \tag{50}$$

Combining (49) and (50) yields

$$\min_{r\in\mathcal{K}}\Big\{\langle x^\star - x, r + A\bar{u}^k - b\rangle - \frac{\|A\|^2}{2\Gamma_k}\|x^0 - x\|^2\Big\} \le \frac{D_{\mathcal{U}}}{\Gamma_k}\left(\frac{\gamma^2_{k+1}}{\tau^2_k} + S_k\right),\quad \forall x \in \mathbb{R}^p. \tag{51}$$

Taking $x := x^0 - \|A\|^{-2}\Gamma_k(A\bar{u}^k - b + r)$ for any $r \in \mathcal{K}$, we obtain from (51) that

$$\min_{r\in\mathcal{K}}\left\{\frac{\Gamma_k}{\|A\|^2}\|A\bar{u}^k + r - b\|^2 + 2\langle A\bar{u}^k - b + r, x^\star - x^0\rangle\right\} \le \frac{2D_{\mathcal{U}}}{\Gamma_k}\left(\frac{\gamma^2_{k+1}}{\tau^2_k} + S_k\right),$$

which implies (by the Cauchy-Schwarz inequality)

$$\min_{r\in\mathcal{K}}\Big\{\Gamma_k\|A\bar{u}^k - b + r\|^2 - 2\|A\|^2\|A\bar{u}^k - b + r\|\|x^\star - x^0\|\Big\} \le \frac{2\|A\|^2D_{\mathcal{U}}}{\Gamma_k}\left(\frac{\gamma^2_{k+1}}{\tau^2_k} + S_k\right).$$

By elementary calculations and $\operatorname{dist}\big(b - A\bar{u}^k, \mathcal{K}\big) = \min\big\{\|A\bar{u}^k - b + r\| : r \in \mathcal{K}\big\}$, we can show from the last inequality that

$$\operatorname{dist}\Big(b - A\bar{u}^k, \mathcal{K}\Big) \le \frac{\|A\|^2}{\Gamma_k}\Big[\|x^0 - x^\star\| + \sqrt{\|x^0 - x^\star\|^2 + \frac{2}{\|A\|^2}\Big(S_k + \frac{\gamma^2_{k+1}}{\tau^2_k}\Big)D_{\mathcal{U}}}\Big]. \tag{52}$$

To prove the first estimate of (37), we use (49) with $x = \mathbf{0}^p$ and $F^\star = -\varphi^\star$ to get

$$\varphi(\bar{u}^k) - \varphi^\star \le \frac{1}{\Gamma_k}\Big[\frac{\|A\|^2}{2}\|x^0\|^2 + \Big(\frac{\gamma^2_{k+1}}{\tau^2_k} + S_k\Big)D_{\mathcal{U}}\Big]. \tag{53}$$

Since we apply Algorithm 1(c) to solve the dual problem (34) using $b_{\mathcal{U}}$ such that $L_b = 1$, we have $S^k \le 2\gamma_1^2$. Then, by using $\gamma_{k+1} = \frac{\bar{c}\gamma_1}{k+\bar{c}}$, and $\tau_k := \frac{1}{k+\bar{c}}$, we can show that $\frac{\gamma^2_{k+1}}{\tau^2_k} = \gamma_1\bar{c}$. Moreover, we also have $\Gamma_k := \sum_{i=0}^k\frac{\gamma_{i+1}}{\tau_i} = \gamma_1\bar{c}(k+1)$. Using these estimates, and $S_k \le 2\gamma_1^2$ from (4) into (52) and (53) we obtain (37). For the left-hand side inequality in the first estimate of (37), we use a simple bound $-\|x^\star\|\operatorname{dist}(b - Au, \mathcal{K}) \le \varphi(u) - \varphi^\star$ for $u = \bar{u}^k \in \mathcal{U}$ from the saddle point theory as in (50). □

## References


1. Andreas Argyriou, Marco Signoretto, and J Suykens. Hybrid conditional gradient-smoothing algorithms with applications to sparse and low rank regularization. *Regularization, Optimization, Kernels, and Support Vector Machines*, pages 53–82, 2014.
2. Michel Baes and Michael Bürgisser. Smoothing techniques for solving semi-definite programs with many constraints. *Optimization Online*, 2009.
3. H.H. Bauschke and P. Combettes. *Convex analysis and monotone operators theory in Hilbert spaces*. Springer-Verlag, 2011.
4. A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding agorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.
5. A. Beck and M. Teboulle. Smoothing and first order methods: A unified framework. *SIAM J. Optim.*, 22(2):557–580, 2012.
6. S. Becker, J. Bobin, and E.J. Candès. NESTA: A fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sci.*, 4(1):1–39, 2011.
7. S. Becker, E. J. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Math. Program. Compt.*, 3(3):165–218, 2011.
8. A. Belloni, V. Chernozhukov, and L. Wang. Square-root LASSO: Pivotal recovery of sparse signals via conic programming. *Biometrika*, 94(4):791–806, 2011.
9. A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: Analysis, algorithms, and engineering applications*, volume 3 of *MPS/SIAM Series on Optimization*. SIAM, 2001.
10. D.P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: Numerical methods*. Prentice Hall, 1989.
11. Radu Ioan Boţ and Christopher Hendrich. A variable smoothing algorithm for solving convex optimization problems. *TOP*, 23(1):124–150, 2012.
12. R.I. Bot and C. Hendrich. A double smoothing technique for solving unconstrained nondifferentiable convex optimization problems. *Compt. Optim. Appl.*, 54(2):239–262, 2013.
13. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
14. J. Chen and S. Burer. A first-order smoothing technique for a class of large-scale linear programs. *SIAM Journal on Optimization*, 24(2):598–620, 2014.
15. P. Combettes and J.-C. Pesquet. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, chapter Proximal Splitting Methods in Signal Processing, pages 185–212. Springer-Velarg, 2011.
16. O. Devolder, F. Glineur, and Y. Nesterov. Double smoothing technique for large-scale linearly constrained convex optimization. *SIAM J. Optim.*, 22(2):702–727, 2012.
17. D. Goldfarb and S. Ma. Fast alternating linearization methods of minimization of the sum of two convex functions. *Math. Program., Ser. A*, 141(1):349–382, 2012.
18. M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. In L. Liberti and N. Maculan, editors, *Global Optimization: From Theory to Implementation*, Nonconvex Optimization and its Applications, pages 155–210. Springer, 2006.
19. I. Necoara and J.A.K. Suykens. Applications of a smoothing technique to decomposition in convex optimization. *IEEE Trans. Automatic control*, 53(11):2674–2679, 2008.
20. V. Nedelcu, I. Necoara, and Q. Tran-Dinh. Computational Complexity of Inexact Gradient Augmented Lagrangian Methods: Application to Constrained MPC. *SIAM J. Optim. Control*, 52(5):3109–3134, 2014.
21. Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Doklady AN SSSR*, 269 (Soviet Math. Dokl.):543–547, 1983.
22. Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.
23. Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optimization*, 16(1):235–249, 2005.
24. Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.
25. Y. Nesterov. Smoothing technique and its applications in semidefinite optimization. *Math. Program.*, 110(2):245–259, 2007.

26. Y. Nesterov. Gradient methods for minimizing composite objective function. *Math. Program.*, 140(1):125–161, 2013.
27. Francesco Orabona, Andreas Argyriou, and Nathan Srebro. PRISMA: Proximal iterative smoothing algorithm. *Tech. Report.*, pages 1–21, 2012. http://arxiv.org/abs/1206.2372.
28. N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.
29. R.T. Rockafellar. *Convexity and Duality in Optimization*, chapter Monotropic Programming: A generalization of linear programming and network programming., pages 10–036. Springer-Verlag, 1985.
30. Q. Tran-Dinh. *Sequential Convex Programming and Decomposition Approaches for Nonlinear Optimization*. PhD Thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, November 2012.
31. Q. Tran-Dinh and V. Cevher. A primal-dual algorithmic framework for constrained convex minimization. *Tech. Report., LIONS*, pages 1–54, 2014.
32. Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. Composite self-concordant minimization. *J. Mach. Learn. Res.*, 15:374–416, 2015.
33. Q. Tran-Dinh, C. Savorgnan, and M. Diehl. Combining Lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems. *Compt. Optim. Appl.*, 55(1):75–111, 2013.
34. J. Yang and Y. Zhang. Alternating direction algorithms for $\ell_1$ -problems in compressive sensing. *SIAM J. Scientific Computing*, 33(1–2):250–278, 2011.